Identify Sparse Vertices

Summary

Identifies route segments with vertices that are farther from the recommended limit in unprojected data.

The output log file contains information about the route segments with sparsely placed vertices. The tool identifies the maximum distance that can be allowed between the vertices for the LRS operations to be performed accurately based on the resolution, the spatial reference and whichever tolerance is smaller, XY or Z.

The formula below determines the maximum length between two vertices on a polyline or polyline segment to successfully have linear referencing calculations performed on the data.

$$L = 2 * \sqrt{R^2 - (R - h)^2}$$

R = the average radius or the lesser of two radii (for example in WGS 1984 R = 6,367,444.657)
h = the smallest of the XY and Z tolerances of the feature class (for example in WGS 1984 H = 0.001 m for the default tolerances)

These values will depend on selected GCS and the XY and Z tolerances configured.

The log file also provides additional information about the number of routes that will contain sparse vertices for various tolerance values**.**

Usage

- The tool works with input data in the geographic coordinate system only.
- The input feature class can be from a file geodatabase or an enterprise geodatabase.
- This tool also accepts other nonLRS line feature classes as input.

Parameters

Label	Explanation	Data Type
Input Feature Class	Input Routes (can either be an LRS network or source routes)	Feature Class
Output Workspace	The enterprise or file geodatabase in which the output feature class will be created. This workspace must already exist.	Workspace
Output Feature Name	The name of the output feature class which will contain the route segments with a smaller number of vertices along with the attributes of the original feature	Feature Class

Derived Output

Label	Explanation	Data Type
Output Feature Class	The output feature class which will contain the route segments with a smaller number of vertices	Feature Class
Output log file	A text file which will list the number and percentage of the routes containing route segments with sparsely placed vertices.	Text File

Code Sample

IdentifySparseVertices example 1 (Python)

The following Python code sample demonstrates how to use the IdentifySparseVertices script.

```
# Name: Identifysparsevertices.py

# Description: Identifies route segments with a smaller number of vertices in unprojected data in a stand-alone script.


# Import arcpy module
```

```
import arcpy


# Import the toolbox
arcpy.ImportToolbox(r"C:\Toolbox\lrs-gcs-toolsets\LRS GCS Toolset.pyt")


# Tool variables
in_features=r"C:\Data\UPDM.gdb\LRS\P_StationSeriesNetwork"
out_workspace=r"C:\Data\UPDM.gdb",
out_feature_name="Routes_Sparsevertices"


# Execute the tool
arcpy.LRSGCSToolset.IdentifySparseVertices(in_features,out_workspace,out_feature_name)
```

IdentifySparseVertices example 2 (stand-alone script)

The following Python code sample demonstrates how to use the IdentifySparseVertices script.

```
# Name: Identifysparsevertices.py
# Description: Identifies route segments with a smaller number of vertices in unprojected data in the python window.


# Import the toolbox
arcpy.ImportToolbox(r"C:\Toolbox\lrs-gcs-toolsets\LRS GCS Toolset.pyt")


# Tool variables
in_features="P_StationSeriesNetwork"
out_workspace=r"C:\Data\UPDM.gdb",
out_feature_name="Routes_Sparsevertices"


# Execute the tool
arcpy.LRSGCSToolset.IdentifySparseVertices(in_features,out_workspace,out_feature_name)
```

Licensing Information

- Basic: No
- Standard: Requires ArcGIS Location Referencing
- Advanced: Requires ArcGIS Location Referencing